

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

METODY POŘÍZENÍ A ZPRACOVÁNÍ OBRAZŮ ZALOŽENÉ NA
ŘÍDKÝCH REPREZENTACÍCH

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ONDŘEJ TALÁR

BRNO 2014



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH
TECHNologiÍ**
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

METODY POŘÍZENÍ A ZPRACOVÁNÍ OBRAZŮ ZALOŽENÉ NA ŘÍDKÝCH REPREZENTACÍCH

**METHODS OF ACQUISITION AND PROCESSING OF IMAGES BASED ON SPARSE
REPRESENTATIONS**

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

ONDŘEJ TALÁR

VEDOUCÍ PRÁCE
SUPERVISOR

Mgr. PAVEL RAJMIC, Ph.D.

BRNO 2014



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Bakalářská práce

bakalářský studijní obor
Teleinformatika

Student: Ondřej Talár

ID: 148143

Ročník: 3

Akademický rok: 2013/2014

NÁZEV TÉMATU:

Metody pořízení a zpracování obrazů založené na řídkých reprezentacích

POKYNY PRO VYPRACOVÁNÍ:

Nastudujte základy problematiky tzv. řídkých reprezentací signálů z doporučených českých i anglických zdrojů. Zaměřte se na oblast zpracování obrazů. Po dohodě s vedoucím práce implementujte v Matlabu úlohu dekvantizace a odšumování a ověřte jejich funkčnost na simulovaných a reálných datech.

DOPORUČENÁ LITERATURA:

[1] Hrbáček, R., Rajmic, P., Veselý, V., Špiřík, J. Řídké reprezentace signálů: komprimované snímání, Elektrevue, 2011. ISSN 1213-1539

[2] Hrbáček, R., Rajmic, P., Veselý, V., Špiřík, J. Řídké reprezentace signálů: úvod do problematiky, Elektrevue, 2011. ISSN 1213-1539

[3] Špiřík, J., Rajmic, P., Veselý, V. Reprezentace signálů: od bází k framům, Elektrevue, 2010. ISSN 1213-1539

Termín zadání: 10.2.2014

Termín odevzdání: 4.6.2014

Vedoucí práce: Mgr. Pavel Rajmic, Ph.D.

Konzultanti bakalářské práce:

doc. Ing. Jiří Mišurec, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Práce se zabývá rekonstrukčními možnostmi, které poskytuje řídká reprezentace signálů. Tato reprezentace zmenšuje signál na pouhý vektor prvků, které ukazují na části signálu ve slovníkové matici. Je nastíněna problematika s kvantovaným signálem a připomenuty druhy modulací, při kterých dochází ke kvantizaci a její způsoby. K řešení je zvolen algoritmus Douglas-Rachford, který nám umožňuje přibližovat se postupně k množině všech přijatelných řešení. Na konci je předvedeno řešení problému a několik testů pro ukázkou funkce vytvořeného programu.

KLÍČOVÁ SLOVA

Řídké reprezentace, dekvantizace, odšumování, dopředně-zpětný algoritmus, Douglas-Rachford algoritmus, rekonstrukce signálu

ABSTRACT

Thesis deals with the reconstruction possibilities provided by the sparse representation of signals. This representation reduces the signal to a mere vector of elements which indicate the signal portion in the dictionary array. It outlined the problems with the quantized signal and recalled modulation type, involving means of quantization. The problem is solved by using Douglas-Rachford algorithm that allows us to approximate on to the set of all acceptable solutions. In the end is demonstrated problem solution and several tests for presentation of created program.

KEYWORDS

Sparse representation, dequantization, denoising, forward-backward algorithm, Douglas-Rachford algorithm, signal reconstruction

TALÁR, Ondřej *Metody pořízení a zpracování obrazů založené na řídkých reprezentacích*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2014. 39 s. Vedoucí práce byl Mgr. Pavel Rajmic, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Metody pořízení a zpracování obrazů založené na řídkých reprezentacích“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce panu Mgr. Pavlu Rajmicovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

Brno

.....

(podpis autora)

PODĚKOVÁNÍ

Výzkum popsáný v této bakalářské práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

Brno

.....
(podpis autora)

OBSAH

Úvod	9
1 Řídké reprezentace signálů	10
1.1 Výchozí znalosti a značení	10
1.2 Báze a framy	10
1.3 Získávání řídkých souřadnic	11
2 Způsoby kvantizace	14
2.1 Pulsně-kódová modulace – PCM	14
2.2 Delta modulace – DM	15
2.3 Adaptivní delta modulace – ADM	15
2.4 Diferenční pulsně-kódová modulace – DPCM	16
2.5 Adaptivní diferenční pulsně-kódová modulace – ADPCM	16
3 Základní úloha	18
3.1 Zadání	18
3.2 Neomezený tvar	19
4 Teoretické řešení	21
4.1 Algoritmy	21
4.2 Proximální operátory	23
5 Praktické řešení	25
5.1 Projekce	25
5.2 Programové řešení	28
6 Testování	33
6.1 Dekvantizace	33
6.2 Odšumování	34
7 Závěr	36
Literatura	37
Seznam příloh	38
A Přílohy na CD	39

SEZNAM OBRÁZKŮ

2.1	PCM modulace s vyznačenými kvantizačními stupni a vlastní kvantizací[6].	14
5.1	Ukázka projektovací funkce pomocí kvadratického programování. . . .	26
5.2	Ukázka rekonstrukce signálu s krokem 2 a 0,5.	30
5.3	Ukázka kvantizace signálu s krokem 2 a 0,5.	31
5.4	Ukázka odchylek signálu pro krok 2 a 0,5.	32
6.1	Výsledky testů dekvantizace signálu pro krok 2.	33
6.2	Výsledky testů dekvantizace signálu pro krok 1.	34
6.3	Výsledky testů odšumování signálu pro krok 1.	35

ÚVOD

Práce pojednává o využití řídkých reprezentací signálů pro konkrétní aplikaci ve zpracování signálů. Konkrétně se jedná o zredukování zkreslení signálu tzv. kvantizačním šumem. Jsou popsány základní poznatky z řídkých signálů, metod a vlastností kvantizace a taktéž je nastíněna metoda řešení problému.

Jedním z důvodů zpracování tématu je seznámení čtenáře se základy řídkých reprezentací signálu, jelikož toto odvětví je poměrně mladé, perspektivní a rychle se rozvíjející. Zjednodušeně lze o těchto signálech říci, že se jedná o nedourčený systém lineárních rovnic, u kterých lze pomocí správného algoritmu nalézt takové řešení, které má mezi všemi nejvíce nulových neznámých. Taková aplikace přináší logické výhody, třeba velkou míru komprese nebo relativně jednoduché zjištění změn v signálu v různých podobách (přítomnost šumu nebo například poškození části signálu). Nevýhodou je potřeba nalezení správné báze signálu, tedy určit, kde je signál řídký. Možností je však více, kupříkladu použít obecnou bázi nebo ji sestavit „na míru“. Také algoritmy pro nalezení vlastního řídkého řešení jsou časově náročné, představují přínos (mimo jiné) hlavně v metodách komprimovaného snímání [3].

V první části práce bude čtenář seznámen se základy řídkého zpracování signálu, budou vysvětleny obecně používané pojmy a taktéž obvyklé metody nalezení řídké reprezentace. V druhé části si připomeneme základní znalosti z kvantizace, včetně několika používaných druhů a definujeme základní problém, kterým se budeme zabývat. Část třetí bude již zaměřena na částečné řešení problému především jeho přeformulováním do podoby potřebné pro jeho širší specifikaci. Po této transformaci bude v části čtvrté nastíněno nejprve obecné řešení a potřebné proměnné k její aplikaci a potom vlastní návrh řešení zadaného problému. V páté části uvedeme praktický návrh programu, včetně ukázky jeho funkčnosti. Šestá část pak obsahuje dosažené výsledky na testovacích vzorcích, zhodnocení dekvantovacích a odšumovacích vlastností programu.

1 ŘÍDKÉ REPREZENTACE SIGNÁLŮ

1.1 Výchozí znalosti a značení

V práci budeme vektory označovat tučnými malými písmeny, jako \mathbf{x} , \mathbf{y} , \mathbf{z} , skalární veličiny budou označené kurzívou, např. x , y , z . Vektory budeme používat primárně sloupcové a pouze s konečným počtem prvků. Indexování prvků vektoru začíná jedničkou, tedy $\mathbf{x} = [x_1, \dots, x_n]$. Matice (báze) označujeme tučnými velkými písmeny, tzn. \mathbf{A} , \mathbf{B} , vektorové prostory označíme jako $\mathbb{R}^n, \mathbb{C}^n$ atd.

Definujeme si také ℓ_p -normu vektoru $\mathbf{x} \in \mathbb{C}^n$ jako:

$$\begin{aligned}\|\mathbf{x}\|_p &= \left(\sum_{i=1}^N |x_i|^p \right)^{1/p} && \text{pro } 1 \leq p < \infty, \\ \|\mathbf{x}\|_p &= \sum_{i=1}^N |x_i|^p && \text{pro } 0 < p < 1, \\ \|\mathbf{x}\|_\infty &= \max_i |x_i|, \\ \|\mathbf{x}\|_0 &= |\text{sup}(x)|.\end{aligned}$$

O normu se striktně vzato jedná pouze v prvním případě, ale pro zjednodušení bude pro všechna p použito označení ℓ_p -norma.

Jedničková norma $\|\cdot\|_1$ představuje součet absolutních hodnot prvků vektoru, norma $\|\cdot\|_0$ pak počet nenulových složek vektoru. Zajímavá je „nekonečná“ norma $\|\cdot\|_\infty$, která nám říká, který prvek z vektoru má nejvyšší hodnotu.

Dále si definujme řídkost vektoru. Řídkost vyjadřuje počet nenulových složek vektoru, můžeme ji tedy zapsat jako:

$$\|\mathbf{x}\|_0 \leq k.$$

Reálné signály ovšem nebývají řídké v pravém slova smyslu podle definic výše, místo nulových složek obsahují malé nenulové hodnoty. Proto se obvykle ještě definuje chyba aproximace, která ale pro naše účely není podstatná. Za zmínku snad stojí fakt, že chyba aproximace se dá v podstatě vyjádřit jako p -normu vektoru, který vznikne z vektoru \mathbf{x} odstraněním k složek s největší velikostí. [2] Chyba tedy vždy závisí na konkrétním vektoru.

1.2 Báze a framy

V následující sekci si objasníme pojem báze, neboli slovník, ve kterém budeme později vyhledávat řídkou reprezentaci signálu. Budeme pracovat s vektorovým prostorem (dále jen VP), jehož prvky jsou vektory. Dimenze VP bude konečné číslo $0 < n \in \mathbb{N}$, množinu konečných vektorů pak bude chápat jako matici, která vznikne naskládáním jednotlivých vektorů do sloupců. [1]

Báze VP je tedy množina lineárně nezávislých vektorů, jejichž lineární kombinací lze vytvořit libovolný vektor v uvažovaném VP. Pokud má dimenze konečný rozměr, pak báze je každá množina obsahující nějaký počet lineárně nezávislých vektorů. [1] Pro ukázkou, pokud máme bázi $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_n]$, pak jestliže hledáme jakýkoliv prvek v naší dimenzi, např. prvek $\mathbf{g} \in \mathbb{V}$, můžeme ho vyjádřit pomocí jednoznačných souřadnic v této bázi jako

$$\mathbf{g} = \sum_{i=1}^n c_i \mathbf{a}_i,$$

kde c_i jsou souřadnice. Pokud má jeden VP více bází, pak všechny jsou tvořeny stejným počtem vektorů.

Nejčastěji používané báze jsou obvykle ortogonální a ortonormální [1]. Pro ortogonální báze platí, že všechny dvojice bázových vektorů jsou na sebe kolmé, ortonormální báze navíc splňují $\|\mathbf{a}_i\|_2 = 1$. Obdobné podmínky platí i pro dvě různé báze ve stejném VP, tedy například matice \mathbf{B} a \mathbf{C} jsou tzv. biortogonální, pokud každý vektor z \mathbf{B} je kolmý na všechny vektory z \mathbf{C} (a obráceně) vyjma jediného. V praxi se obvykle používají báze (bi)ortogonální [3], z důvodů lepší numerické stability [1].

Zajímavou formou báze je frame. Jak již bylo řečeno, bázi v konečněrozměrném prostoru tvoří lineárně nezávislé vektory, jejichž počet je stejný jako dimenze VP. Pokud ale z nějakého důvodu přidáme do báze další prvky, budou již nutně lineárně závislé. Takto upravenou bázi nazýváme pak frame nebo konečný frame. Mohlo by se zdát, že tyto vektory navíc nemají žádné opodstatnění, ale opak je pravdou. Nadbytečnost vyjádření může být i výhodou. Frame je méně omezený než báze a proto se používá pro svou flexibilitu, nevýhodou však jsou náročnější výpočetní metody a riziko numerické nestability [1]. Dále se proto framy nebudeme zabývat, v naší úloze si plně vystačíme s klasickou ortonormální bází.

1.3 Získávání řídkých souřadnic

Známe už všechny potřebné pojmy, jak ale nalezneme řídké řešení signálu? Může se na první pohled zdát, že se jedná o jednoduchou úlohu. Řešíme prakticky obvyklou soustavu lineárních rovnic $\mathbf{A}\mathbf{x} = \mathbf{y}$, ovšem s tím, že neznámý a hledaný vektor \mathbf{x} má být co nejřidší, tedy má obsahovat co nejvíce nulových složek. Úloha se dá zapsat zapsat takto:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{vzhledem k} \quad \mathbf{A}\mathbf{x} = \mathbf{y} \quad (1.1)$$

kde známe vektor $\mathbf{y} \in \mathbb{C}^m$, což je náš pozorovaný nebo měřený signál. Taktéž známe matici $\mathbf{A} \in \mathbb{C}^{m \times N}$. Budeme předpokládat pouze případy $m < N$, resp. $m \ll N$, a \mathbf{A} je plně řádkové hodnosti [2].

Matici \mathbf{A} bude dále označovat jako slovník, sloupce matice jsou pak atomy. Všechna \mathbf{x} , která splňují $\mathbf{Ax} = \mathbf{y}$, nazýváme přípustná řešení nebo přípustné reprezentace vektoru \mathbf{y} . Je známo, že za dosavadních podmínek je nekonečně mnoho, proto tedy hledáme specifické požadavky na slovník, které zaručí možnost „rekonstrukce“ původního \mathbf{x} z \mathbf{y} [2].

Představme si tedy libovolnou soustavu lineárních rovnic, která má jediné řešení s řádkostí 1. Pokud tedy změním koeficienty této soustavy, povede to k tomu, že budou existovat 2-řádká nebo 3-řádká řešení. Tato úvaha vede k aproximační úloze. Povolí se totiž malá odchylka od požadavku:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{vzhledem k} \quad \|\mathbf{Ax} - \mathbf{y}\|_p \leq \alpha, \quad (1.2)$$

kde nejčastěji uvažujeme $p = 2$. Tento typ úlohy se obvykle používá v případech, kdy je signál zašumněn [2].

Pro jednoznačnost řešení si musíme zavést určité podmínky. Jedním z nich je číslo $\text{spark}(\mathbf{A})$, které nám říká, jaký je nejmenší počet lineárně nezávislých sloupců matice:

$$\text{spark}(\mathbf{A}) = \min_{\mathbf{z} \in \mathbf{A}, \mathbf{z} \neq 0} \|\mathbf{z}\|_0.$$

U nenulové matice $\mathbf{A} \in \mathbb{C}^{m \times N}$, kde $m < N$, platí, že spark může nabývat hodnot $\text{spark}(\mathbf{A}) \in \{2, \dots, m+1\}$. Čím menší spark je, tím řidší musí vektor \mathbf{x} být, aby byla zajištěna jedinečnost tohoto řešení. Bohužel, nalezení $\text{spark}(\mathbf{A})$ je srovnatelně výpočetně náročné jako řešení problému (1.1).

Důležitou součástí při hledání řešení se stává vzájemná koherence, která je definována jako největší absolutní normalizovaný skalární součin dvou různých sloupců matice,

$$\mu(\mathbf{A}) = \max_{1 \leq j, k \leq N, j \neq k} \frac{|\mathbf{a}_j \mathbf{a}_k|}{\|\mathbf{a}_j\|_2 \cdot \|\mathbf{a}_k\|_2},$$

kde \mathbf{a}_j označujeme j -tý sloupec matice \mathbf{A} . Pomocí vzájemné koherence můžeme zjistit „míru lineární závislosti“ mezi sloupci [2]. Naší snahou je hledat takové matice, které jsou co nejvíce nekoherentní, neboť to nám zajišťuje jedinečnost řešení. Jestliže jsme řekli, že budeme používat ortonormální slovníky, očekáváme tedy koherenci minimální.

Předpokládejme tedy nyní, že jsme splnili podmínky pro jednoznačnost řešení. Pokud však trváme na přesném řešení, je nutné projít všechny kombinace podmnožin sloupců matice. To ovšem v praxi, kdy máme velkou matici, není přijatelné. Výsledek v rozumném čase je ale možné získat aproximační úlohou.

Hlavním problémem normy ℓ_0 je ten, že není konvexní funkcí a tudíž nelze používat pro řešení problému (1.1) známé metody. Víme ale, že normy ℓ_p jsou konvexní

pro $p \geq 1$. Proto se pro hledání řídkých signálů používá častěji aproximační úloha:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{vzhledem k} \quad \mathbf{Ax} = \mathbf{y} \quad (1.3)$$

V mnoha případech je řešení (1.1) a (1.3) shodné, což je pro naši práci velice příznivé. Doplníme, že v případě zašumněných dat je úloha obdobná k (1.2), tedy:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{vzhledem k} \quad \|\mathbf{Ax} - \mathbf{y}\|_2 \leq \alpha \quad (1.4)$$

Podmínky, za kterých se řešení obou minimalizací shodují, jsou stále předmětem hledání. Může se totiž stát, že při hledání řešení podle (1.3) najdeme ne jedno nebo dvě řešení, ale prakticky nekonečné množství řešení, přičemž ale jen ty „krajní“ budou ty očekávané. Pro úplnost si tedy naznačíme pár známých podmínek, které zajišťují jedinečnost řešení.

První z podmínek je tzv. Null Space Property (NSP). Nebudeme uvádět přesné definice, uvedeme pouze zjednodušenou verzi. Jedná se o podmínku, jejíž splnění nám říká, že norma je koncentrována v „malém“ počtu prvků. Zajišťuje též, že řešení je jednoznačné a lze ho nalézt ℓ_1 -minimalizací. Zároveň také platí, že pokud je možné ze soustavy $\mathbf{Ax} = \mathbf{y}$ získat všechny řídké vektory \mathbf{x} , pak matice \mathbf{A} splňuje NSP [2].

Druhou z podmínek je tzv. Restricted Isometry Property (RIP). Nabízí výpočetně přijatelnější alternativu, protože je stabilní i pod vlivem šumu. Zde hlavně uvedeme matice, které této podmínce vyhovují. Jedná se o Gaussovske a Bernoulliho matice specifických tvarů [2]. Tyto matice mají jednu podstatnou výhodu, a to tu, že jejich rekonstrukční schopnosti jsou invariantní vůči unitární matici, což je výhoda například u komprimovaného snímání [3].

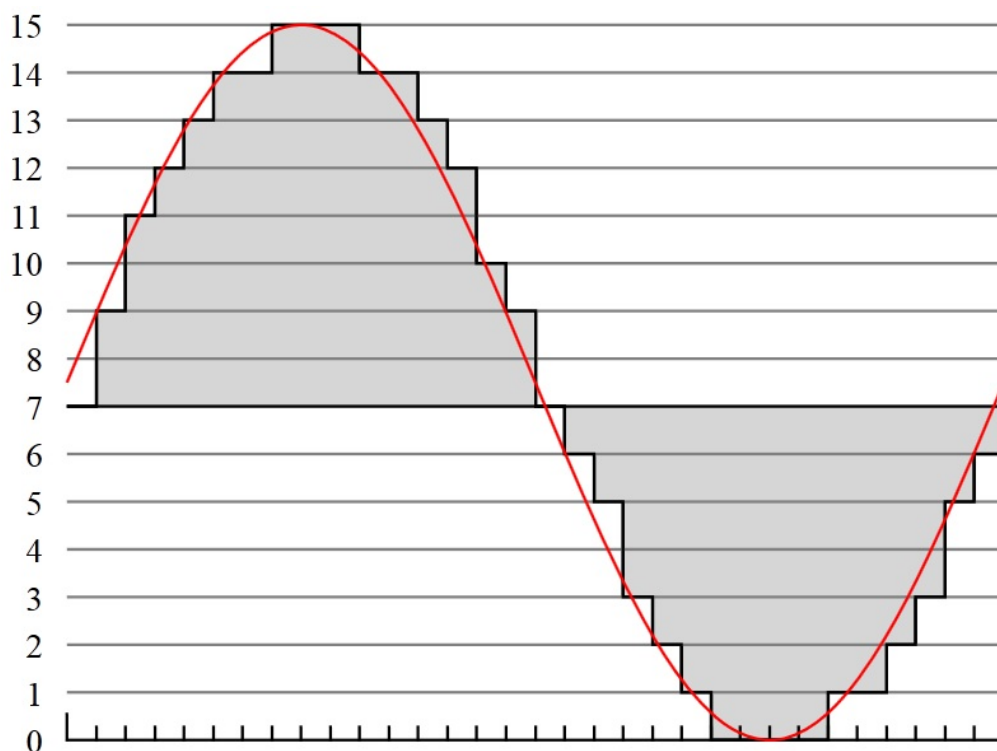
Pro hledání řídkých reprezentací se využívá několika algoritmů, které se dají rozdělit do dvou skupin. První skupina metod vychází přímo z ℓ_1 -relaxace, proto se nazývají relaxační. Tyto algoritmy spoléhají na to, že se za určitých podmínek dostaneme k přesnému nebo relativně přesnému řešení. Druhou skupinu tvoří tzv. greedy algoritmy, jejichž princip spočívá v tom, že při každém průchodu najedou jeden nebo více nejvýznamnějších atomů a tyto mají dále podíl na dalším řešení. Mimo tyto kategorie stojí algoritmy založené na prahování (thresholding) nebo takové, které využívají vlastností obou skupin [2].

2 ZPŮSOBY KVANTIZACE

2.1 Pulsně-kódová modulace – PCM

Pulsně-kódová modulace (dále jen PCM) je základní metoda digitální reprezentace a přenosu analogových signálů. Tvoří standard pro digitální audio v počítačích, digitální přenos v telefonii a dalších aplikacích. U PCM rozhodují o zpětné rekonstrukci původního signálu dva faktory: kmitočet vzorkování, tedy kolik vzorků signálu je pořízeno (čím více, tím lepší je rekonstrukce) a míra kvantizace, tedy mezi kolik kvantizačních stupňů je signál zakódován. Opět platí, že čím více stupňů, tím vyšší je přesnost rekonstrukce.

Nás bude především zajímat postup kvantizace. Představme si tedy nějaký navzorkovaný signál. Na tento signál aplikujeme kvantizaci, která má tolik stupňů, na kolik bitů se kóduje (pro názornost např. 3 bity, což je 8 možností). Jednotlivé vzorky signálu jsou tedy posunuty na jednu z těchto kvantizačních hladin. Mezi hladinami je uprostřed rozhodovací úroveň; pokud je vzorek pod touto úrovní, kóduje se na nižší kvantizační stupeň, pokud je nad úrovní, přesune se na vyšší stupeň. Tím se však logicky dopouštíme chyby, která je sice částečně kompenzována v demodulátoru, ovšem stále zde zůstává tzv. kvantizační zkreslení [5].



Obr. 2.1: PCM modulace s vyznačenými kvantizačními stupni a vlastní kvantizací[6].

2.2 Delta modulace – DM

Základní princip modulace delta spočívá v porovnávání signálu v komparátoru s generátorem sledovacího průběhu. Do komparátoru vstupuje spojitý vstupní signál a stupňovitý sledovací signál. Tento je vytvářen v generátoru sledovacího průběhu. Komparátor v pravidelných časových okamžicích porovnává oba signály. Pokud je vstupní signál větší hodnoty než sledující, je vyslán impuls a sledovací signál je o jeden stupeň zvýšen. Pokud je vstupní signál menší než sledovací, není vyslán žádný impuls a sledovací průběh o stupeň klesne. Informaci o stoupání nebo klesání nese jediný bit, modulace se tedy někdy nazývá jednobitová modulace delta.

Při přenosu kódový převaděč vypustí záporné impulsy a unipolární sled kladných impulsů a mezer přemění na bipolární, vhodný pro přenos. Na straně přijímače kódový převaděč obnoví unipolární signál, v případě příchodu jedničky propustí měnič polarity impulsů impulsy v nezměněné podobě, v případě příchodu nuly se změní polarity na opačnou. Velkou důležitostí má dolní propust, která ze stupňovitého průběhu vytvoří signál shodný s původním signálem, tedy až na kvantizační zkreslení.

Výhodou modulace delta oproti jiným typům digitálních modulací je jednodušší obvodové uspořádání, nevýhodou je zejména potřeba vyššího kmitočtu impulsů a potíže s omezováním výšky přenášeného impulsu. Při přenosu vyšších kmitočtů se totiž může stát, že sledovací průběh začne klesat dříve, než dosáhne potřebnou výšku [5].

2.3 Adaptivní delta modulace – ADM

Adaptivní modulace delta se liší od klasické jednobitové delta modulace tím, že velikost jednotlivých stupňů není konstantní, ale mění se v závislosti na průběhu modulovaného signálu, mluvíme o adaptivní kvantizaci. Výška vstupního signálu je omezována méně než u klasické DM. Nejjednodušší řešení spočívá v tom, že pokud je vstupní signál větší než sledovací, je každý následující stupeň jeden a půl násobkem předchozího, pokud je menší než sledovací, je každý následující stupeň polovinou předchozího. Kvantizační kroky se mohou měnit jak již bylo uvedeno, nebo v každém druhém kroku, s konstantním rozdílem sousedních kvantizačních kroků nebo se kvantizační kroky mohou měnit spojitě.

Napětí na integrační kapacitě integrátoru je závislé na výstupní posloupnosti kodéru. Kvadrátor převádí napětí různé polarity na kondenzátoru na stejnosměrné řídicí napětí pro řízený zesilovač a umožňuje také nelineární změnu velikosti kvantizačního kroku. Princip je obvodově jednoduchý, nevýhodou je nutnost shodnosti charakteristik kodéru a dekodéru. Tyto nedostatky odstraňuje např. ADM s digitálním řízením.

Důležitým blokem ADM s digitálním řízením je detektor přetížení. Pokud výstupní posloupnost kodéru obsahuje tři symboly stejné polarity, což svědčí o přetížení kodéru, objeví se na výstupu detektoru přetížení napětí. Toto napětí je po sečtení s pomocným napětím minimálního kvantizačního kroku převedeno na pomocný integrátor, což vede ke zvětšení velikosti kvantizačního kroku. Čím častěji se na vstupu detektoru objevují trojice impulsů stejné polarity, tím větší je velikost kvantizačního kroku. Neobsahuje-li výstupní posloupnost kodéru tři symboly stejné polarity, je na výstupu detektoru přetížení nula a velikost kvantizačního kroku je dána pomocným napětím[5].

2.4 Diferenční pulsně-kódová modulace – DPCM

U delta modulace nese informaci o změně signálu jeden bit, u diferenční pulsně-kódové modulace (dále jen DPCM) se kóduje pomocí PCM rozdíl mezi skutečnou a předpokládanou velikostí vzorku. Tento předpoklad lze uskutečnit s ohledem na znalost libovolnou znalost předchozích vzorků, v nejjednodušším případě se spokojíme s porovnáváním dvou sousedních vzorků. Pravděpodobně platí, že u většiny reálných signálů lze velikost vzorku odhadnout se znalostí předchozích vzorků a rozdíl mezi skutečnou a předpokládanou hodnotou bude menší než celý vzorek. Proto je modulací DPCM většinou kvantováno jemněji než modulací PCM.

Předpokládaný signál je tedy vlastně součet předchozích vzorků vstupního signálu zvětšené o nějakou reálnou konstantu a uvažujeme pouze lineární predikci. Pro optimální hodnotu predikovaného signálu bude střední kvadratická hodnota rozdílů vstupního a predikovaného signálu minimální [5].

2.5 Adaptivní diferenční pulsně-kódová modulace – ADPCM

Podívejme se nyní na signál pro jednoduchost jako na řečový. DPCM je ideální pro kódování konstatního výkonu (hlasitosti) s korelační funkcí stanovenou na základě dlouhodobého výkonu. Ve skutečnosti se výkon i korelace mění v závislosti na druhu hlasu, hlasitosti i skladbě řeči. ADPCM přizpůsobuje kvantování zejména velkým změnám výkonu kódovaného signálu, popřípadě způsobuje kvantování navíc krátkodobým změnám hustoty rozdělení okamžitých hodnot signálu. Rozdílové vzorky vznikají stejně jako u DPCM. Podle jejich efektivních hodnot je měněna v řízeném zesilovači jejich výška, následuje pevná kvantizace. Užívá se též možnost proměnné kvantizace s různou šířkou kvantizačních stupňů kvantizátoru při neřízené výšce rozdílových vzorků.

Zisk řízeného zesilovače lze často měnit v závislosti na efektivní hodnotě rozdílových vzorků pouze skokem. Přepínané a zpětné převodníky PAM/PCM mají k dispozici např. čtyři různé kompresní a jim odpovídající expanzní charakteristiky, které jsou přepínány podle okamžité hodnoty hustoty rozdělení rozdílových vzorků.

Hlavní výhoda ADPCM je především na poli telefonie. Má totiž tu vlastnost, že při stejném kvantizačním zkreslení (tím pádem stejné kvalitě přenosu) je možné snížit přenosovou rychlost jednoho zakódovaného hovoru na polovinu, tedy snížit počet bitů, kterými je kódováno, z osmi na čtyři. Tím pádem je možné přenést dvojnásobný počet hovorů za stejnou jednotku času[5].

3 ZÁKLADNÍ ÚLOHA

3.1 Zadání

Specifikujme si tedy náš problém. Mějme signál \mathbf{z} , který je diskrétní (tedy již navzorkovaný) a nakvantovaný. Kvantován byl s konstantním krokem, např. jako PCM modulace. Tento signál je blíže nespecifikované délky, kvantován je taktéž na prozatím nespecifikovaný počet stupňů. Pozorovaný signál tedy označme jako $Q^\alpha(\mathbf{z})$, kde α je vzdálenost mezi jednotlivými kvantizačními stupni. Rozhodovací úroveň označíme $\alpha/2$, tedy v polovině stupně se rozhoduje o tom, na kterou úroveň se vzorek signálu posune.

Náš problém je následující: jak co nejvíce potlačit kvantizační zkreslení? Řekněme si tedy ještě jednu důležitou věc. Víme, že původní nenakvantovaný signál \mathbf{z} má nějakou řídkou reprezentaci \mathbf{x} , opět blíže nespecifikované řídkosti, v nějakém slovníku \mathbf{A} . Můžeme tedy napsat, že $\mathbf{z} = \mathbf{A}\mathbf{x}$. Řešení ovšem není jednoduché, neboť toto platí pro původní signál, který hledáme. Náš signál byl již zatížen kvantizačním šumem, který je třeba odstranit. Hledáme tedy takovou řídkou reprezentaci, která se bude co nejvíce blížit původnímu nezašumněnému signálu. Jak ale zvolit maximální přijatelnou chybu? Víme, že při kvantování se složky vzorkovaného signálu posunují na příslušné kvantovací stupně. O tom, kam se posunou, rozhoduje rozhodovací úroveň, přesně v polovině mezi nimi. Můžeme tedy říci, že pokud bude námi rekonstruovaná složka signálu ležet někde mezi kvantizačním stupněm a rozhodovací hladinou, pak toto je tolerovatelná chyba. Aby rekonstrukce byla zároveň i efektivní, budeme požadovat současně co nejmenší vektor řídkosti \mathbf{x} . Matematicky tedy řešíme optimalizační problém, daný rovnicí

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{vzhledem k} \quad \|Q^\alpha(\mathbf{z}) - \mathbf{A}\mathbf{x}\|_\infty \leq \alpha/2. \quad (3.1)$$

Pro vysvětlení, proč používáme jako prvek porovnání nekonečnou normu. Tento typ normy nám vrátí největší prvek vektoru. Pokud odečteme od nakvantovaného signálu náš odhad, tak jako výsledek dostaneme prvek nejvyšší hodnoty. Pokud nejvyšší prvek bude splňovat nerovnost, pak logicky i ostatní prvky vektoru musejí nerovnost splňovat. Průkaznost a správnost řešení je tak silně ošetřena proti možným chybám.

Řešení tohoto našeho problému však není tak jednoznačné, jak by se na první pohled mohlo zdát. V této formě se hledání vhodných prvků může stát velice zdlouhavým procesem. První úvaha je tedy jednoznačná: dala by námi specifikovaná úloha transformovat na jiný typ úlohy, který nám ale i přesto dá stejný nebo alespoň velmi podobný výsledek? Pro odpověď je třeba sáhnout např. k metodám tzv. blízkého rozdělávání (proximal splitting)[4].

3.2 Neomezený tvar

Rozhodli jsme se tedy použít nějakou z metod blízkého rozdělování. Naše základní úloha ale není uzpůsobena k těmto řešením, musíme ji tedy transformovat. Největší zádrhel způsobuje omezenost základní úlohy. Vyčleňujeme pouze část, kde máme nějaké řešení, ale nevíme, co se stane, když se dostaneme mimo rozsah přijatelných řešení. Naštěstí se mnoho optimalizačních problémů dá zapsat v tzv. neomezeném tvaru (unconstrained form)[4], který v základním tvaru vypadá takto

$$\min_{\mathbf{x} \in \mathbb{R}^N} f_1(\mathbf{x}) + \dots + f_m(\mathbf{x}),$$

kde f_1, \dots, f_m jsou konvexní funkce od dimenze \mathbb{R}^N do $(-\infty, +\infty)$. Problém může vyvstat z vlastního faktu, že mnoho funkcí není diferencovatelných, což vyřazuje množství konvenčních metod snadných optimalizačních technik [4].

Aplikujme nyní tuto základní neomezenou formu na rovnici. Budeme hledat tvar zhruba v tomto smyslu

$$\min_{\mathbf{x}} \lambda \|\mathbf{x}\|_1 + \frac{1}{2} \|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2.$$

Obě funkce jsou pevně vymezeny. Naše λ bude jedničkové, neboť prozatím s vektorem řídkosti nechceme provádět žádné změny. Potíže ale může dělat druhá funkce. Jak víme, naše kontrolní složka má poměrně specifický tvar. Naštěstí existuje pro nás výhodná indikátorová funkce [4]. Její definice je v podstatě prostá: mějme C , které je neprázdnou podmnožinou dimenze \mathbb{R}^N . Indikátorová funkce C je potom

$$i_C : \mathbf{x} \mapsto \begin{cases} 0 & \text{pokud } \mathbf{x} \in C, \\ +\infty & \text{pokud } \mathbf{x} \notin C. \end{cases}$$

Pro názornost si podstatnou vlastnost této funkce ukážeme na příkladě. Mějme tedy za úkol minimalizovat například funkci jako

$$\min_{\mathbf{y} \in \mathbb{R}^N} i_C(\mathbf{y}) + \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2.$$

Zde je jasně vidět důležitost a krása indikátorové funkce. Pokud nějaký prvek \mathbf{y} bude prvek z dimenze, ve které pracujeme, a zároveň bude patřit do stanovené funkce C , pak indikátorová funkce bude nulová a řešíme pouze druhou část rovnice. Jestliže ale prvek neleží v námi stanovených mezích uvnitř dimenze, funkce roste k nekonečnu a druhá část rovnice je zanedbatelná. Minimalizovat tento nekonečný případ by eventuálně šlo, ale pro nás je v tuto chvíli důležité, že jsme našli prvek, který nevyhovuje požadavku, kladený mezemi a je nutno hledat jiný prvek.

Jak jsme si ukázali, indikátorová funkce nám tedy říká, kdy je prvek v námi připuštěných mezích. Tato vlastnost nám velmi pomůže při převodu zbytku rovnice

na neomezený tvar. Jednoduše si řekněme: co je naše omezovací funkce? Odpověď je jasná, celá druhá část rovnice, kde se ptáme, jestli je největší prvek v mezích rozhodovací úrovně. Řečeno jinak, zavedeme, že $C = \{x \mid \|Q^\alpha(\mathbf{z}) - \mathbf{A}\mathbf{x}\|_\infty \leq \alpha/2\}$. S tímto zjednodušením máme hotový převod na neomezený tvar. Zkrátka budeme hledat řešení rovnice

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 + i_C. \quad (3.2)$$

Máme zaručen jak hledaný vektor, tak omezení, že se budeme pohybovat pouze v mezích a nikde jinde. Bude však ještě potřeba dalších transformací pro účely vlastního algoritmu, ale nejtěžší část máme za sebou.

4 TEORETICKÉ ŘEŠENÍ

4.1 Algoritmy

Pro vyřešení našeho minimalizačního problému použijeme algoritmy z rodiny tzv. dopředně-zpětného rozdělování (forward-backward splitting) [4]. Určeme si tedy dvě funkce, z nichž jedna je konvexní a diferencovatelná pomocí tzv. Lipschitzova spojitého gradientu ∇f_2 [4], tedy

$$\left(\forall (x, y) \in \mathbb{R}^N \times \mathbb{R}^N \right) \quad \|\nabla f_2(y) - \nabla f_2(x)\| \leq \beta \|x - y\|$$

kde $\beta \in (0, +\infty)$. Předpokládejme, že obě naše funkce jsou kladné, stejně jako prvky řídkosti, které hledáme. Pak můžeme tedy říci, že hledáme řešení problému

$$\min_{x \in \mathbb{R}^N} f_1(x) + f_2(x).$$

Z výše uvedených podmínek lze zjistit [4], že existuje alespoň jedno přípustné řešení, které lze pro nějaké $\gamma \in (0, +\infty)$ charakterizovat rovnicí

$$x = \text{prox}_{\gamma f_1}(x - \gamma \nabla f_2(x)),$$

která naznačuje možnost opakování. Když upravíme rovnici vzhledem k této možnosti, pak

$$x_{n+1} = \text{prox}_{\gamma_n f_1}(x_n - \gamma_n \nabla f_2(x_n)),$$

přičemž uvažujeme pouze takové hodnoty krokovacího parametru γ_n , které vyhovují zadaným mezím. Toto je základní schema dopředně-zpětných algoritmů, které se dá rozdělit na část dopřednou (forward), používající funkci f_2 , a na část zpětnou (backward), využívající funkci f_1 [4]. Za poznámku stojí fakt, že pokud $f_1 = 0$, tak se bavíme o gradientní metodě [4]

$$x_{n+1} = x_n - \gamma_n \nabla f_2(x_n),$$

která slouží k minimalizaci funkce s Lipschitzovým spojitým gradientem. Na druhou stranu, pokud $f_2 = 0$, pak se rovnice změní v algoritmus blízkého bodu (proximal point algorithm) [4]

$$x_{n+1} = \text{prox}_{\gamma_n f_1},$$

který slouží k minimalizaci nediferencovatelných funkcí. Dopředně-zpětné algoritmy tedy kombinují obě tyto části, což má za následek postupné přibližování minimalizačních bodů do námi určených mezí. Nejzákladnější forma algoritmu z řady zpětně-dopředných s použitím uvolňovacího parametru λ_n tedy vypadá takto[4]:

Určíme $\epsilon \in (0, \min 1, 1/\beta), x_0 \in \mathbb{R}^N$

Pro $n = 0, 1, \dots$

$\gamma_n \in \langle \epsilon, 2/\beta - \epsilon \rangle$

$y_n = x_n - \gamma_n \nabla f_2(x_n)$

$\lambda_n \in \langle \epsilon, 1 \rangle$

$x_{n+1} = x_n + \lambda_n (\text{prox}_{\gamma_n f_1} y_n - x_n)$

Je dokázáno [4], že každá sekvence $(x_n)_{n \in \mathbb{N}}$ konverguje k řešení základní rovnice. Tato forma algoritmu používá různou velikost kroku, ale relaxační parametr nemůže přesáhnout jedničku. Z tohoto základu vychází další úpravy, například můžeme použít konstatní přibližovací krok a zvětšit uvolňovací parametr.

Z výše uvedených definic ale vyplývá, že pro použití dopředně-zpětného algoritmu musí být alespoň jedna z funkcí diferencovatelná pomocí Lipschitzova spojitého gradientu. Zkusme ale tuto podmínku taktéž uvolnit a říct si, že nám stačí jakákoliv jiná podmínka, která nám ale zaručí, že obě dvě funkce nemusejí být diferencovatelné. Taková podmínka se vskutku najít dá a příslušný algoritmus se nazývá Douglas-Rachford [4]. Podle něj, náš základní problém připouští alespoň jedno řešení a toto řešení je pro nějaké $\gamma \in (0, +\infty)$ chakterizováno pomocí dvou podmínek:

$$x = \text{prox}_{\gamma f_2} y$$

$$\text{prox}_{\gamma f_2} y = \text{prox}_{\gamma f_1} (2\text{prox}_{\gamma f_2} y - y)$$

Pokud tedy opravdu požadujeme minimalizaci dvou nediferencovatelných funkcí a je možné splnit uvedené podmínky, tento algoritmus nabízí velmi zajímavou možnost realizace. Vlastní algoritmus vypadá obecně takto:

Určíme $\epsilon \in (0, 1), \gamma > 0, y_0 \in \mathbb{R}^N$

Pro $n = 0, 1, \dots$

$x_n = \text{prox}_{\gamma f_2} y_n$

$\lambda_n \in \langle \epsilon, 2 - \epsilon \rangle$

$y_{n+1} = y_n + \lambda_n (\text{prox}_{\gamma f_1} (2x_n - y_n) - x_n)$

Stejně jako dopředně-zpětné algoritmy využívá Douglas-Rachford rozdělování díky tomu, že postupně používá nejprve první a poté druhou funkci odděleně. Lze na

něj tedy pohlížet jako na zobecnění dopředně-zpětných případů, jelikož nevyžaduje Lipschitzův spojitý gradient. Na druhou stranu je ale nutno tuto absenci vyvažovat numericky, což místo jednoho přibližovacího kroku vynucuje přidání ještě druhého kroku pro zajištění stability. Rozdíl mezi dopředně-zpětným algoritmem a Douglas-Rachfordem však není příliš jasný, oba dovedou být velmi efektivní[4].

4.2 Proximální operátory

V algoritmech se často vyskytují zkratky pro proximální operátory[4]. Jejich využití je velmi výhodné a důležité, proto si je teď zkráceně definujeme. Zavedme si tedy problém

$$\min_{\mathbf{y} \in \mathbb{R}^N} f(\mathbf{y}) + \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2,$$

který pro každé $\mathbf{x} \in \mathbb{R}^N$ připouští jedinečné řešení, které lze označit jako $\text{prox}_f \mathbf{x}$. Operátor prox_f nyní definovaný je tedy proximální operátor funkce [4]. Proximální operátor si můžeme charakterizovat například ještě takto

$$\left(\forall (x, p) \in \mathbb{R}^N \times \mathbb{R}^N \right) \quad p = \text{prox}_f x \quad \Leftrightarrow \quad x - p = \nabla f(p),$$

kde je funkce f diferencovatelná. Jak již bylo řečeno, proximální operátory mají velmi atraktivní vlastnosti, které je činí vhodnými pro iterativní minimalizační algoritmy [4]. Z hlediska zpracování signálů mají proximální operátory velmi přirozenou interpretaci ve smyslu odšumování, což velmi pomůže při řešení našeho problému, který do této kategorie spadá.

Nejprve je však nutno najít si správné reprezentace našich proměnných, aby bylo možné je zakomponovat do algoritmu. Nejprve si převedeme první část rovnice. Jak víme, $\|\mathbf{x}\|_1$ je v podstatě součet absolutních hodnot prvků vektoru. Hledejme tedy nějakou funkci, která bude mít vlastnosti absolutní hodnoty. Jednu takovou funkci vskutku nacházíme [4]:

$$\sigma_{|\underline{\omega}, \bar{\omega}|}(x) = \begin{cases} \underline{\omega}x & \text{pokud } x < 0, \\ 0 & \text{pokud } x = 0, \\ \bar{\omega}x & \text{jinak.} \end{cases} \quad (4.1)$$

Pokud si za horní a dolní mez dosadíme, například $\bar{\omega} = 1$ a $\underline{\omega} = -1$, zjistíme, že tato funkce přesně odpovídá funkci absolutní hodnoty. Proximální operátor k této funkci je pak tento:

$$\text{soft}_{|\underline{\omega}, \bar{\omega}|}(x) = \begin{cases} x - \underline{\omega} & \text{pokud } x < \underline{\omega}, \\ 0 & \text{pokud } x \in [\underline{\omega}, \bar{\omega}], \\ x - \bar{\omega} & \text{pokud } x > \bar{\omega}. \end{cases} \quad (4.2)$$

Operátor soft zde značí tzv. měkké prahování (soft thresholding). Jak je tedy vidět, v tomto případě je absolutní hodnota v jistém smyslu ekvivalentem měkkému prahování. Pro druhou část rovnice bude hledání jednodušší. Máme zde naši indikátorovou funkci i_C , která nám omezuje meze hledání. Pro ni platí, že její proximální operátor je projekce $P_C \mathbf{x}$. Pro lepší představu, projekce $P_C \mathbf{x}$ nějakého $\mathbf{x} \in \mathbb{R}^N$ vůči neprázdné uzavřené konvexní množině $C \subset \mathbb{R}^N$ je řešení rovnice

$$\min_{\mathbf{y} \in \mathbb{R}^N} i_C(\mathbf{y}) + \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2.$$

S pomocí těchto všech uvedených změn můžeme sestavit základní verzi algoritmu, který vyřeší náš původní problém. Jeho tělo bude Douglas-Rachford algoritmus, pouze doplníme naše zjištěné parametry:

Určíme $\epsilon \in (0, 1), \gamma > 0, \mathbf{y}_0 \in \mathbb{R}^N$

Pro $n = 0, 1, \dots$

$$\mathbf{x}_n = P_{\gamma C} \mathbf{y}_n$$

$$\lambda_n \in \langle \epsilon, 2 - \epsilon \rangle$$

$$\mathbf{y}_{n+1} = \mathbf{y}_n + \lambda_n \left(\gamma \text{soft}_{|\underline{\omega}, \bar{\omega}|} (2\mathbf{x}_n - \mathbf{y}_n) - \mathbf{x}_n \right).$$

5 PRAKTICKÉ ŘEŠENÍ

5.1 Projekce

Pozornost nyní věnujme vlastní projekci, která je z celého algoritmu nejtěžší na realizaci. Jak již bylo řečeno, jedná se o proximální operátor druhé funkce, tedy té, která nám zásadně vymezuje prostor přijatelných řešení. Pokud si tedy rozepíšeme proximální operátor projekce, pak hledáme:

$$P_{\{x \mid \|Q^{\alpha}(\mathbf{z}) - \mathbf{Ax}\|_{\infty} \leq \alpha/2\}} \mathbf{y}$$

Nabízí se otázka, jak tedy prakticky omezit podmínky? Pro názornost se nyní omezíme jen na 2D prostor. Představme si, že je zde část tohoto prostoru vyhovující našim řešením. Je však potřeba ji ohraničit, například nějakým geometrickým tvarem. Pokud si zvolíme například trojúhelník, pak můžeme lehce popsat omezující podmínky pomocí rovnic přímk, které tvoří jeho strany. Tyto rovnice se pak dají sepsat do matice, jako příklad zde uvedeme rovnice pro pravoúhlý trojúhelník:

$$\begin{bmatrix} -1 & 0 \\ 0 & -1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Nyní se s použitím správné programovací funkce budou body projektovat na úzce vymezenou část celého prostoru. Podobně se dají sestavit tvary i pro vícerozměrný prostor, ale to prozatím pomineme. V naší situaci se jeví jako výhodná konstrukce čtverec se středem v nule a stranami délky α , tedy od nuly k jedné straně je to $\alpha/2$. Díky tomu splníme základní podmínku projekce, tedy že body se musí projektovat do prostoru maximálně o $\alpha/2$ od původního bodu.

Než se však pustíme dále, tak pro testovací účel nahradíme vektor výsledků jedničkami. Pro úplnost, takto vypadá základní matice pro ukázkou projektovací funkce:

$$\begin{bmatrix} -1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

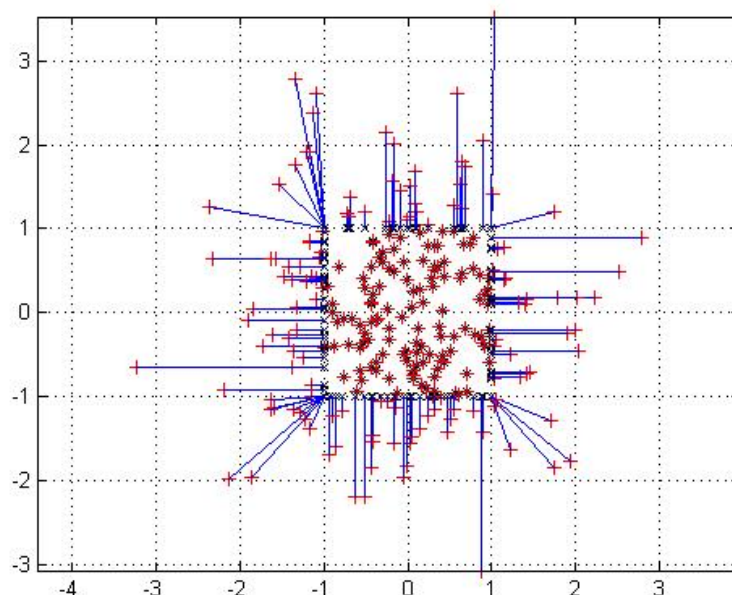
Vlastní projekci zajistíme pomocí tzv. kvadratického programování (quadratic programming). Využijeme pro to funkci `quadprog` z knihovny MATLABu. Tato optimalizační funkce řeší problém typu

$$\min_{\mathbf{x}} \frac{1}{2} \mathbf{x}^T \mathbf{H} \mathbf{x} + \mathbf{f}^T \mathbf{x} \text{ tak, že } \mathbf{Ax} \leq \mathbf{b}.$$

Některé matice a vektory známe, například známe hledaný vektor \mathbf{x} nebo slovníkovou matici \mathbf{A} , ale především nás zajímají zbývající složky této funkce. Pro jejich zjištění si musíme rozepsat, co vlastně hledáme. Naším úkolem je zmenšovat rozdíl dvou vektorů pomocí iterací, abychom došli postupně k výsledku. Jejich rozdíl můžeme z ℓ_2 -normy přepsat na mnohem uchopitelnější tvar pro kvadratické programování. Po roznásobení tedy zjistíme, že

$$\|\mathbf{x} - \mathbf{x}_0\|^2 = (\mathbf{x} - \mathbf{x}_0)^T (\mathbf{x} - \mathbf{x}_0) = \mathbf{x}^T \mathbf{x} - 2\mathbf{x}_0^T \mathbf{x} + \mathbf{x}_0^T \mathbf{x}_0.$$

Jen krátkým pohledem zjistíme, že nyní už známe další proměnné do kvadratického programování. Matice \mathbf{H} bude jednotková, ale musí být vynásobena dvěma, abychom se zbavili zlomku z rovnice pro kvadratické programování. Za vektor \mathbf{f} pak dosadíme člen $-2\mathbf{x}_0$, kde \mathbf{x}_0 představuje vektor bodů, které budeme projektovat. Ještě jsme nezmínili vektor \mathbf{b} , ten je však jasný, je to vektor výsledků. Pokud se podíváme zpět na popisující matici, tak jsou to samé jedničky.



Obr. 5.1: Ukázka projektovací funkce pomocí kvadratického programování.

Že takto funkce opravdu funguje se můžeme přesvědčit na obrázku 5.1. Je jasně patrné, že body z okolí se projektují nejkratší možnou cestou na kraje čtverce, a body, které již leží uvnitř obrazce, zůstávají na svém místě, jelikož splňují podmínku.

Bohužel, jakkoliv by se tento tvar zdál ideální, není tomu tak, protože zde vstupuje ještě jeden prvek, tedy matice \mathbf{A} . Ta nám náš čtverec zdeformuje, ale díky

tomu bude možné správně rekonstruovat body. Opět si pro příklad tuto matici zjednodušíme na jakoukoliv náhodnou matici. Pokud spolu vynásobíme matici čtverce a náhodnou matici, zjistíme pár zajímavých věcí, které s výhodou využijeme. Především lze zjistit, že jakkoliv původní obrazec zdeformuje, jeho střed zůstává na stejném místě. Je to logické, nula zůstává nulou vždy, ale platí to i pro pozdější aplikace. Dále je významný fakt, že i při deformaci zůstává při lineárním zobrazení zachována rovnoběžnost. Tuto vlastnost opět oceníme, nemusíme řešit problémy s proměnou čtverce v náhodný čtyřúhelník.

Nabízí se možná otázka, jak vyřešit změnu vektoru výsledků. Pokud změněme hlavní matici, tak i výsledky se musejí náležitě změnit. Tuto změnu ale můžeme snadno pokrýt. Místo složitěho řešení a následného dělení výsledků můžeme vynásobit matici čtverce náhodnou maticí, ale inverzní. Řešení získáme v požadované formě, tedy s vektorem výsledků nezměněným.

Máme pokrytu tedy základní myšlenku projekce. Vytvoříme kolem určitého bodu čtverec, ten zdeformujeme další maticí na tento tvar projektujeme náhodný bod. Nyní si ale ukážeme, že až tolik náhody jak v bodu, tak v matici nebude. Jako první musíme opět zmínit fakt, že vektor výsledků nebudou v praxi jedničky, nýbrž poloviny kvantovacích kroků. Chceme totiž získat bod, jehož maximální odchylka je právě půlka kvantovací hladiny. Musíme však navíc vyřešit zásadní otázku, tedy jak posunou čtverec se středem v nule kamkoliv do prostoru. Zde je odpověď prostá, zkrátka k vektoru výsledků přičteme souřadnice požadovaného bodu, v tomto případě celého signálu. Je také třeba dbát na správné znaménko, abychom neměnili podmínky v rovnicích. Pro ilustraci, jestliže budeme mít vektor \mathbf{b} reprezentující signál, pak matice bez deformace pomocí \mathbf{A} vypadá takto:

$$\begin{bmatrix} -1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} \alpha/2 - b_1 \\ \alpha/2 + b_1 \\ \alpha/2 - b_2 \\ \alpha/2 + b_2 \end{bmatrix}$$

Dosud jsme uvažovali matici, která nám upraví původní čtverec, jako náhodnou. Náhodná ale nebude, chceme totiž vytvořit rekonstrukci signálu, který vytvoříme pomocí částí slovníku. Tento slovník zároveň použijeme jako deformační matici, abychom získávali adekvátní změnu a získali řídký vektor \mathbf{x} .

Vytvářet matice a vektory dle výše uvedené ukázky by ale bylo příliš složité a zdržovalo by chod programu. Když se podíváme na matici čtverce, tak vždy se pod sebou budou střídát kladné a záporné jedničky, stejně tak se na straně výsledků se bod buď přičte nebo odečte. Tato zajímavost se dá dobře využít, protože seřazením kladných a záporných prvků k sobě vniknou dvě jednotkové matice, jedna kladná a druhá záporná.

Můžeme si tuto situaci ilustrovat již jen obecně:

$$\begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix} \cdot \mathbf{Ax} \leq \begin{bmatrix} \alpha/2 + \mathbf{b} \\ \alpha/2 - \mathbf{b} \end{bmatrix}$$

Formálně můžeme vynásobit matici slovníku s jednotkovou maticí jak kladnou, tak zápornou a získat tím zjednodušení. Nemusíme již přemýšlet nad čtvercem, matice je tvořena pouze z kladného a záporného slovníku. Opět pro ilustraci toho, co jsme udělali za změnu:

$$\begin{bmatrix} \mathbf{A} \\ -\mathbf{A} \end{bmatrix} \cdot \mathbf{x} \leq \begin{bmatrix} \alpha/2 + \mathbf{b} \\ \alpha/2 - \mathbf{b} \end{bmatrix}$$

Tímto je celá projekce vyřešená a stejně tak je použita v programu. Zbývajících část je měkké prahování, ale to je triviální věc. Existuje funkce v MATLABu, která prahování provede, ale my použijeme vlastní funkci, která bude pracovat jen s koeficienty a skalárem pro prahování.

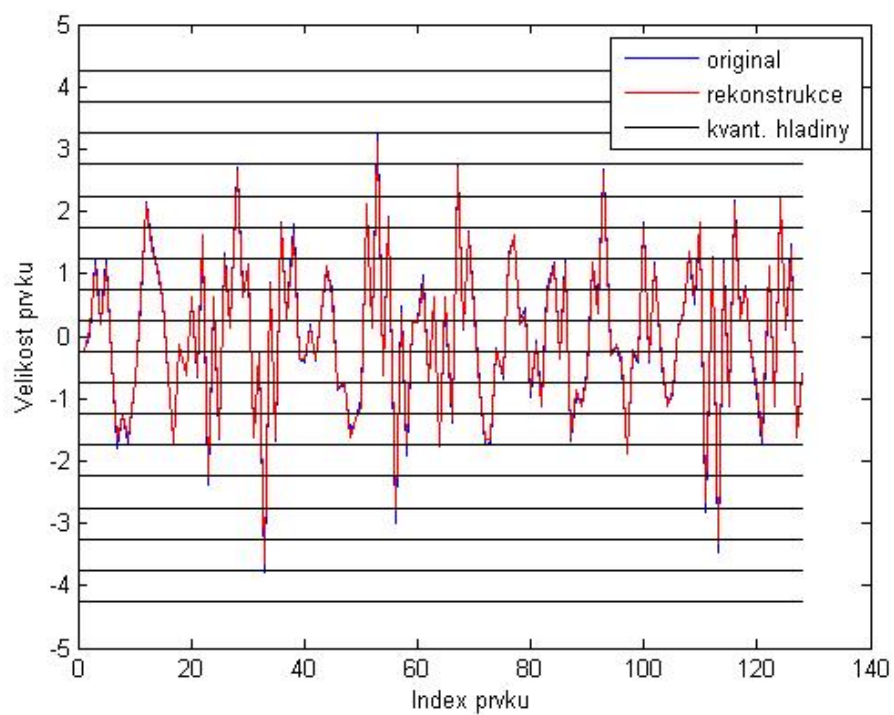
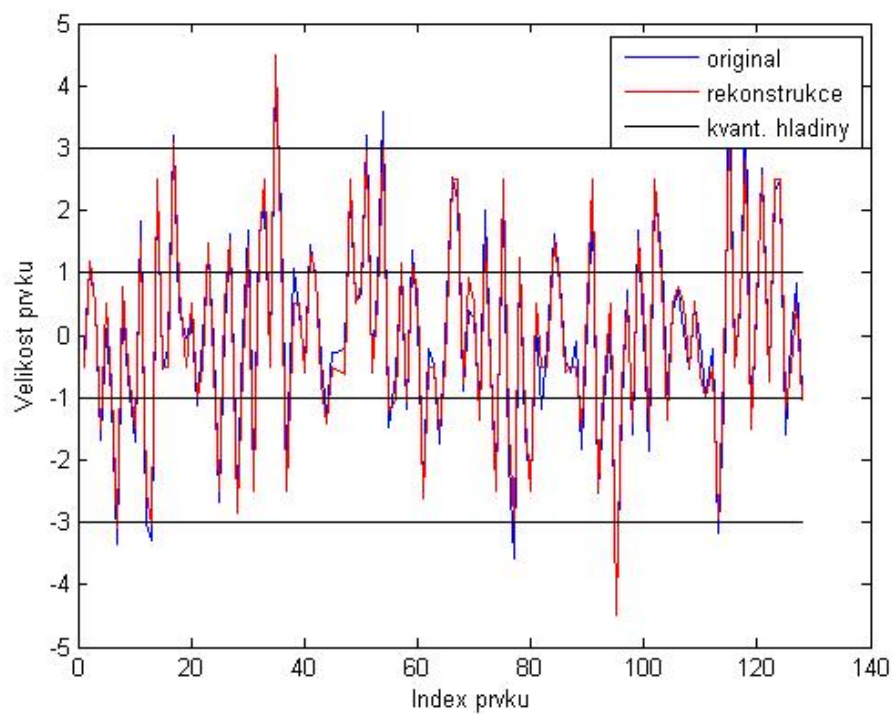
5.2 Programové řešení

Již bylo řečeno, že pro řešení využijeme prostředí MATLAB. Program se v principu skládá z generátoru náhodného signálu, kvantizátoru, projekčních funkcí, funkce měkkého prahování a iterační smyčky. Jako slovník jsme zvolili bázi DCT, ze které uživatel vybere počet sloupců, ty se poté naváhují a sečtou do jednoho signálového vektoru. Kvantizátor pak tento signál nakvantuje v takovém rozsahu a na takové hladiny, které opět zadá uživatel. Dále už pak následují funkce pro přípravu matic a vektorů pro projekci a vlastní projekce se spolu s prahováním provádí ve smyčce. Pro naši ukázkou používáme signál délky 128 s vybráním 10 sloupců.

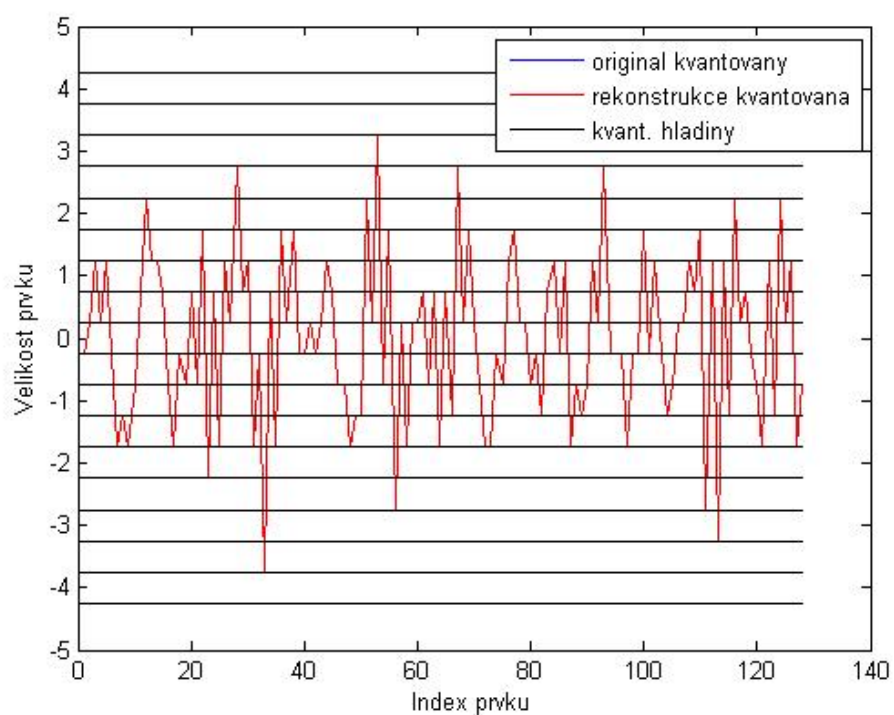
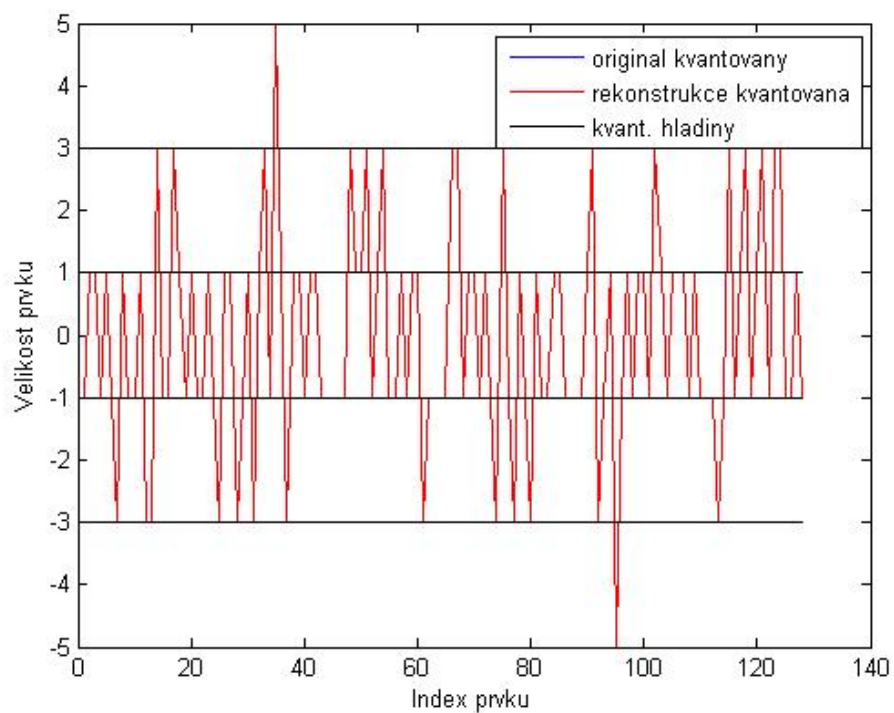
Vzhledem k nepřesné rekonstrukci se nabízí otázka, zda není v nějakém bodě signálu porušena podmínka o tom, že maximální odchylka nemůže být větší než polovina kvantovací hladiny. V části s testováním si ukážeme více, ale pro náš signál, jehož ukáзка je na obrázcích, se odchylka rekonstruovaného signálu za tuto mez nedostane.

Na obrázcích 5.2 je vidět, že metoda není dokonalá. Je možné, že existuje chyba v programu, nicméně v průběhu nebyla nalezena, tedy toto je finální rekonstrukční schopnost. Je možné si povšimnout pochopitelně lepší rekonstrukce při menším odstupu kvantovacích hladin. Pro ověření jsme rekonstruovaný signál opět nakvantovali. Z obrázků 5.3 je jasně patrné, že kvantovaný signál se s rekonstruovaným kvantovaným signálem překrývá. V této rovině je tedy metoda částečně účinná, rekonstruovaný signál má stejnou kvantizaci.

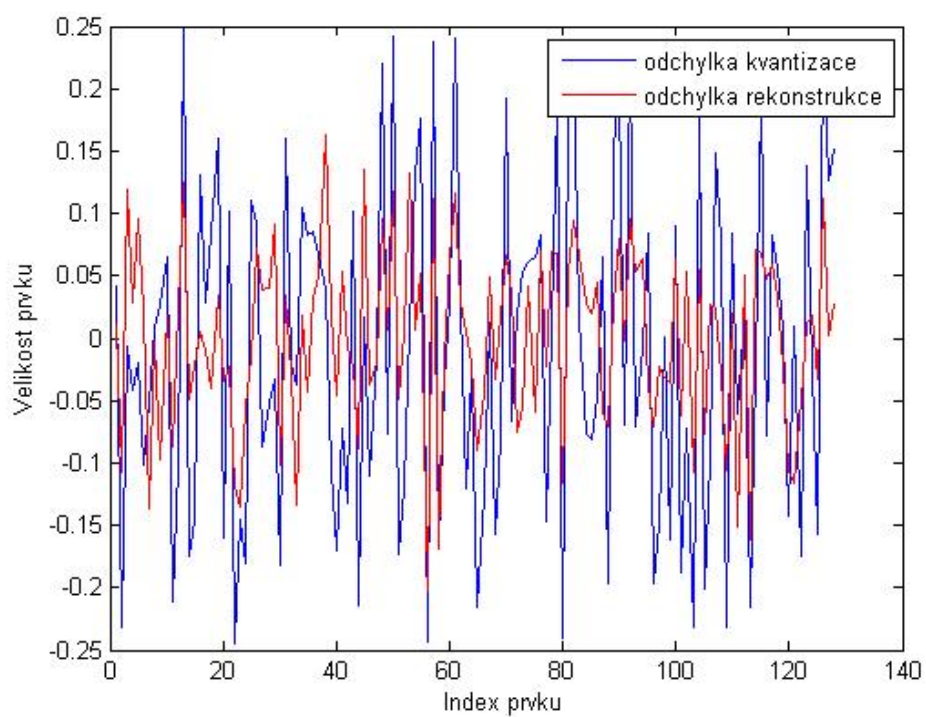
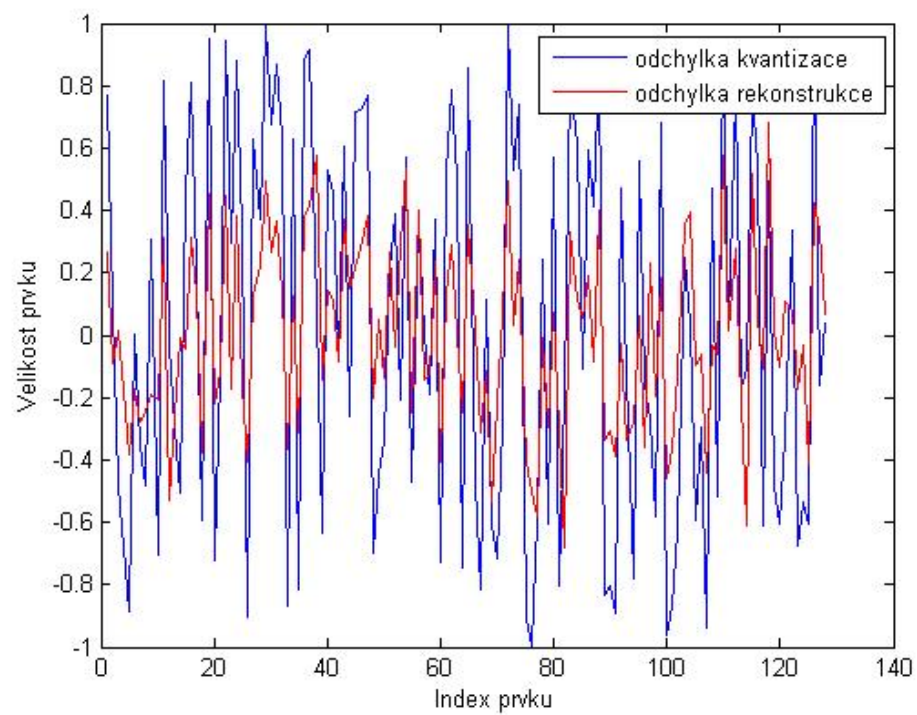
Z obrázků 5.4 vyplývá, že z větší části je odchylka signálu rekonstruovaného menší, než odchylka signálu kvantovaného. Co je důležitější, červená odchylka rekonstrukce se drží pod polovinou kroku i ve svém maximu, což je věc žádoucí. Důvody pro nedokonalou hlavní rekonstrukci jsou tedy stále nejasné, ačkoliv se zdá, že by jí nic nemělo bránit.



Obr. 5.2: Ukázka rekonstrukce signálu s krokem 2 a 0,5.



Obr. 5.3: Ukázka kvantizace signálu s krokem 2 a 0,5.

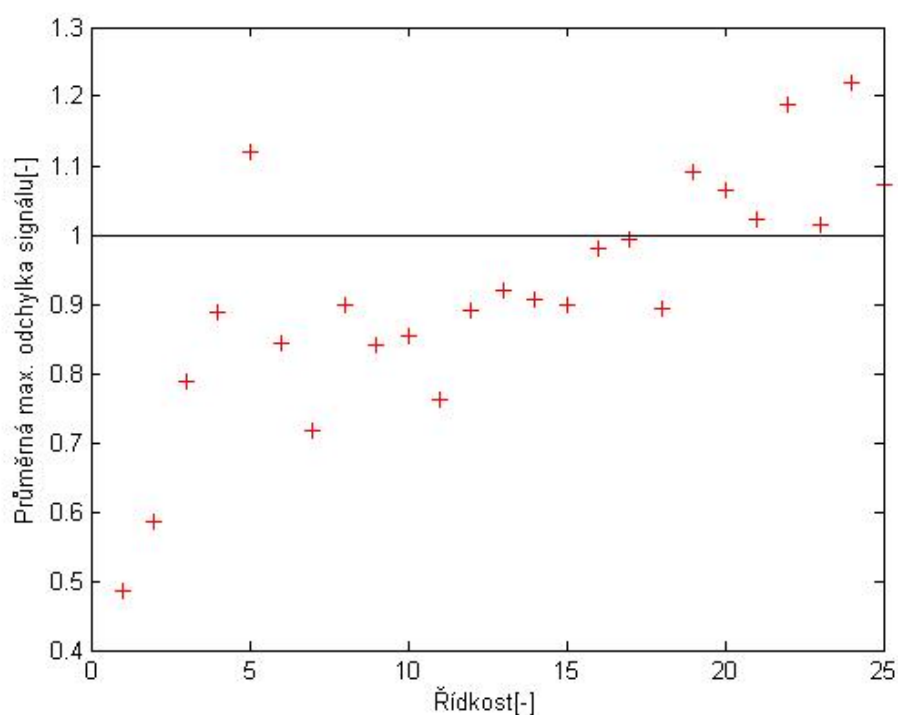


Obr. 5.4: Ukázka odchylek signálu pro krok 2 a 0,5.

6 TESTOVÁNÍ

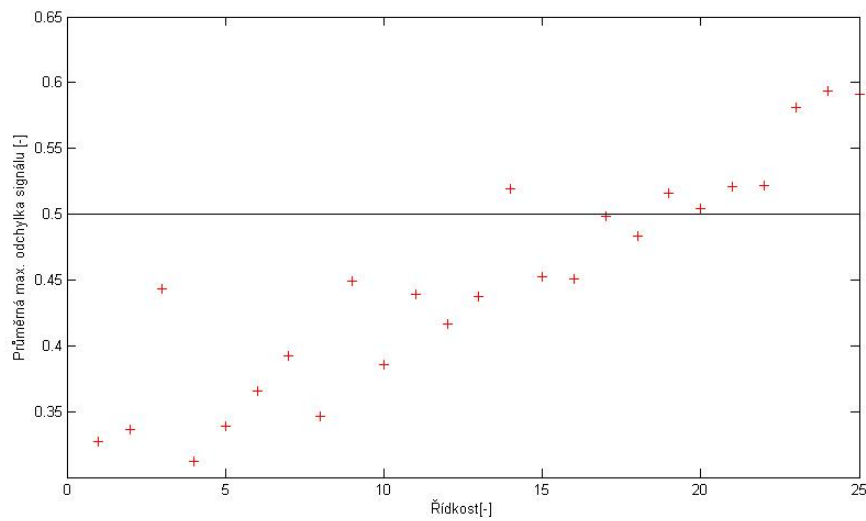
6.1 Dekvantizace

Náš primární cíl spočívá v dekvantizaci, v této části si tedy ukážeme, jak obstál náš algoritmus při větším testování. Opět jsme používali signál délky 128 z báze DCT, nicméně nyní s proměnnou řídkostí, kterou reprezentují vybírané sloupce. Každý řídkost byla otestována desetkrát, chyba je tedy průměrem ze všech testů. Ty jsme prováděli pro různou velikost kroku.



Obr. 6.1: Výsledky testů dekvantizace signálu pro krok 2.

Z grafů 6.1 a 6.2 je patrné, že metoda je značně nestálá, ale je samozřejmé, že v rámci deseti pokusů nemůžeme očekávat nejpřesnější výsledky. Testy jsou ale časově velmi náročné, proto je použito jen deset opakování. Můžeme pozorovat lineární charakter maximální chybovosti, která se ovšem nikdy nedostane nad hranici kvantovacího kroku. Cokoliv nad polovinou nás ale nezajímá, neboť to nenaplnuje náš požadavek na maximální dochylku půl kvantovacího kroku.

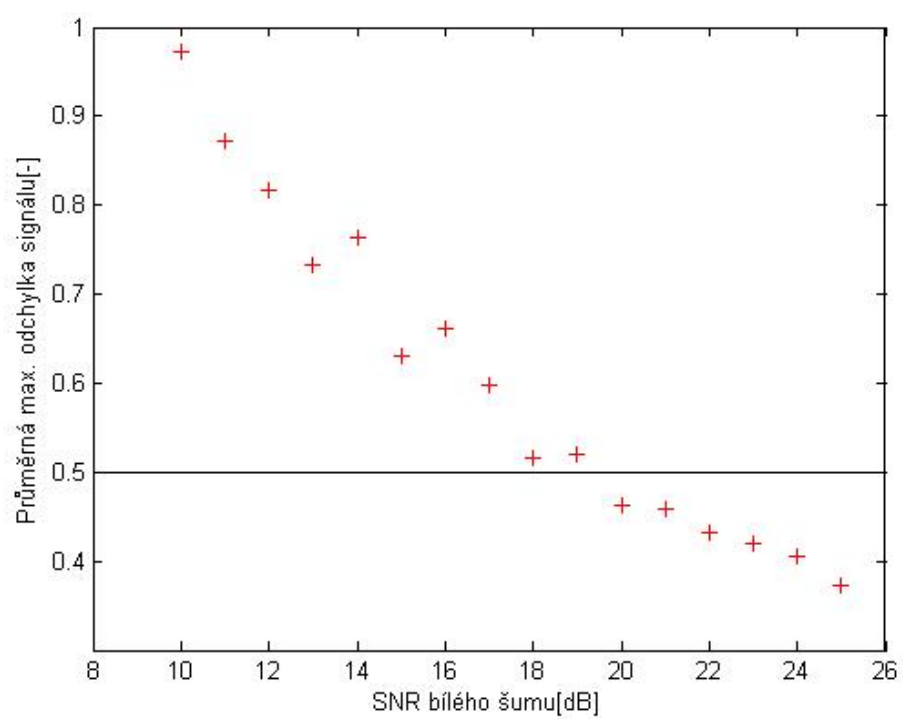


Obr. 6.2: Výsledky testů dekvantizace signálu pro krok 1.

6.2 Odšumování

Při rekonstrukci signálu se vyskytla i otázka, zda bude možné provést rekonstrukci i pro zašumněný signál. Provedli jsme tedy sérii testů s Gaussovým bílým šumem různé síly, který je přičten k signálu před kvantizací a následně podroben rekonstrukčnímu algoritmu. Signál je stále délky 128 z báze DCT, řídkost je pevně stanovena na 7.

Z grafu 6.3 vyplývá, že odstup signál šum musí být alespoň 20 dB, aby byla možná rekonstrukce podle našich podmínek. To sice není ideální, ale opět je to důkazem, že naši aplikaci lze využít i tímto způsobem. Taktéž je zřejmý kvadratický ráz grafu.



Obr. 6.3: Výsledky testů odšumování signálu pro krok 1.

7 ZÁVĚR

V práci jsme teoreticky dokázali, že aplikace řídkých reprezentací signálů pro rekonstrukci je možná a poměrně lákavá možnost vzhledem k silným opravným vlastnostem. Vlastní metody používané ke zjišťování a získávání řídkých souřadnic se neustále vyvíjí, stejně tak se rozvíjejí možnosti jak získat kvalitní a co nejvíce specifický slovník, který nám bude zaručovat ještě kvalitnější výsledek. Naše výsledky s použitím dopředně-zpětného algoritmu ukazují na jistou schopnost rekonstrukce signálu i odšumování, nicméně je znát, že nejsou tak výrazné, jak jsme dle teorie usuzovali. Je velmi pravděpodobné, že se však jedná pouze o chybu v našem programu, nicméně nebyla v průběhu tvorby a testování odhalena. S jistotou však můžeme tvrdit, že rekonstrukce velmi řídkých signálů je ve většině případů možná a docela přesná. Co se zašumněných dat týče, je zde také potenciál pro další využití řídkých signálů. Je však potřeba dalšího výzkumu nad rámec této práce, aby se docílilo ideálního výsledku, jak slibuje teorie.

LITERATURA

- [1] Špiřík, J.; Rajmic, P.; Veselý, V.: *Reprezentace signálů: od bází k framům*. Elektrorevue – internetový časopis, 2010: 10 s.
Dostupné z URL: <http://elektrorevue.cz/cz/download/reprezentace-signalu--od-bazi-k-framum/>.
- [2] Hrbáček, R.; Rajmic, P.; Veselý, V.;aj.: *Řídké reprezentace signálů: úvod do problematiky*. Elektrorevue – internetový časopis, 2011: 10 s.
Dostupné z URL: <http://www.elektrorevue.cz/cz/download/ridke-reprezentace-signalu--uvod-do-problematiky/>.
- [3] Hrbáček, R.; Rajmic, P.; Veselý, V.;aj.: *Řídké reprezentace signálů: komprimované snímání*. Elektrorevue – internetový časopis, 2011: 8 s.
Dostupné z URL: <http://www.elektrorevue.cz/cz/download/ridke-reprezentace-signalu--komprimovane-snimani/>.
- [4] Combettes, L. P.; Pesquet, J-C.: *Proximal splitting methods in signal processing*. Fixed-Point Algorithms for Inverse Problems in Science and Engineering, 2011: 28 s.
Dostupné z URL: <http://arxiv.org/abs/0912.3522v4>.
- [5] ŠKORPIL, Vladislav. *Přístupové a transportní sítě*. Skripta FEKT VUT v Brně. Brno, 2012. 128 s. ISBN 978-80-214-4457-7.
- [6] *PCM modulace s kvantizací*. Dostupné z URL: http://www.cem.brighton.ac.uk/staff/alb14/CI221/Lectures%20&%20Tutorials/Lecture_3_Audio_Files.html.

SEZNAM PŘÍLOH

A Přílohy na CD

39

A PŘÍLOHY NA CD

Na CD se nachází elektronická verze bakalářské práce a používané programy.

Programy jsou spustitelné v prostředí MATLAB, verze 2011a a vyšší. Archiv obsahuje tyto složky:

1. Základní program – obsahuje základní tělo programu se všemi potřebnými funkcemi, spouští se skriptem „Hlavní program“, uživatel může uvnitř skriptu nastavit svoje hodnoty
2. Test dekvantizace – obsahuje skript „Test dekvantizace“, který spustí dávkové testování; uživatel může opět uvnitř skriptu měnit parametry
3. Test odsumovani – obsahuje skript „Test odsumovani“, který spustí dávkové testování; uživatel může opět uvnitř skriptu měnit parametry